combinatorialOptimizationHW1

0656124 劉承順

1. (a) To finding the minimum T, it suffices to solve the DIRECTED MINIMUM MEAN CYCLE(MMC) PROBLEM of G. the relation is as follows:

(1)If G has MMC, is the minimum cycle mean, the optimum T is .

(2)If G doesn’t have any cycle, T can be arbitrary small(but T is time, ).

Pf:

Consider the MMC of G,

With , we have

constructed in (b) satisfy the constraint for all cycle, so we know

(b) in each connected component of G, the first value can be randomly assigned to a random vertex v. Do a dfs in from v, let

while doing dfs.

1. I use C++ to implement two programs of Dinic’s algorithm.

I use the method in a paper[1] to implement version of dinics(second program), although I didn’t use all method in the paper(use the fact that each network needing a blocking flow is not only acyclic but layered). It is much slower than original version of dinics.

The testdata is BVZ-tsukuba0, which is from <http://vision.csd.uwo.ca/data/maxflow/>

It has 110594 nodes and 514483 arcs. My first program costs about 1 second to run it, and second program runs about twenty minutes. Two programs both get correct answer.

1. For every maximum flow in G, its crossing edges from to and from to are all saturated, because and are minimum cuts.

Edges from to can be classified to (1) to (2) to . Edges in (1) are saturated, because . And edges in (2) are also saturated, because , . so for every maximum flow in G, edges from to are all saturated. And is a minimum cut.

To find a minimum s – t cut with |X| as large as possible, let

1. Convert all edges in G into two edges with . Compute MINIMUM CAPACITY CUT problem between s and t.